Applicants file a Request for Reconsideration succinctly pointing out the elements in the pending claims that have not been addressed, as well as showing how the language of the claims patentably distinguishes from each of the references cited by the examiner. The present paper is submitted in accordance with the Supervisory Examiner's instructions.

CLAIM REJECTIONS UNDER 35 USC §102

Applicants respectfully traverse the rejection of claims 1-27 under 35 U.S.C. 102(e) as being anticipated by *Levine et al.*, U.S. Patent No. 5,835,702 ("Levine"). In order for a claim to be anticipated under 35 USC §102(e), the anticipating reference must teach, suggest or disclose each and every element of the claim. See, for example, C.R. Bard, Inc. v. M3 Systems, 48 U.S.P.Q.2d 1225, 1230 (Fed. Cir. 1998) ("When the defense of lack of novelty is based on a printed publication that is asserted to describe the same invention, a finding of anticipation requires that the publication describe all of the elements of the claims, arranged as in the patented device..."); In re Bond, 15 U.S.P.Q2d 1566, 1567 (Fed. Cir. 1990) ("For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference...").

Applicants respectfully submit the Office Action has not demonstrated that each and every element of claims 1-27 is taught or disclosed by Levine, and therefore has not established a *prima facie* case of anticipation.

General Summary of Levine:

Levine is generally directed to collecting performance data in a performance monitor implemented in the hardware of a processor. Code points instruct the performance monitor to begin collecting raw counter information into control registers of the performance monitor (Levine, 2:50-63). Both the control registers and performance monitor of Levine are implemented in hardware (Levine, Figure 4). Levine's point of novelty is directed to a dependency between a first and a second counting function within his hardware performance monitor, where the first counting function is performed by a particular counter and the first counting function depends on a particular event programmed within the second counter.

With this point of novelty of Levine, reprogramming of all code points for each performance counter is not required (Levine, abstract).

General Summary of the present application:

Unlike Levine, the present application is not directed towards collecting performance data in a hardware performance monitor. Rather, the present application is generally directed to rendering collected performance data using a software-based solution. That is, the present application provides a framework and methods for rendering performance data between counter providers (e.g., applications that collect performance counts) and counter consumers (e.g., applications that use the collected performance counts). With the present application, a counter provider application may have a memory structure within its allocated address space in which to collect performance data. The counter provider may register itself and corresponding memory structure via a set of APIs (application program interfaces) provided by an operating system. Thereafter, a counter consumer application, via additional APIs, may set up a buffer in which to receive counter data collected by the counter provider and may request extraction of the counter data from the counter provider's registered memory structure into the buffer. An embodiment of an architectural view of the framework and methods may be found in the specification, Figure 3.

Independent Claims 1, 9, 17 and 25:

Elements That Have Not Been Addressed:

Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest each and every element of independent claims 1, 9, 17 and 25. For example, Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest at least the elements of independent claim 1: "a processor," "an operating system executed by the processor" and "an application program interface of the operating system, comprising a set of functions." In the Office Action's remarks on page 2 with regard to claim 1, no citations to any passages of Levine for any of the above elements were provided. The Office Action's Response to Arguments on pages 5-6 also did not provide any citations to Levine with regard to any of the above elements of claim 1.

3

Similarly, with regard to independent claim 9, Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest every element of claim 9. For example, the Office Action has not cited to portions of Levine that teach, disclose or suggest at least the element of claim 9 "providing, via the operating system, an application program interface comprising a set of functions, the set of functions comprising a counter registration API function and a counter query API function." In the Office Action's remarks on page 2 with regard to claim 9, no citations to any passages of Levine for any of the above elements were provided. The Office Action's Response to Arguments on pages 5-6 also did not provide any citations to Levine with regard to any of the above elements of claim 9.

Still similarly, Applicants respectfully submit that the Office Action also has not cited to portions of Levine that teach, disclose or suggest every element of independent claim 17. For example, the Office Action has not cited to portions of Levine that teach, disclose or suggest at least the element of claim 17: "providing, via the operating system, an application program interface comprising a set of functions, the set of functions comprising a counter registration API function and a counter query API function." In the Office Action's remarks on page 2 with regard to claim 17, no citations to any passages of Levine for any of the above elements were provided. The Office Action's Response to Arguments on pages 5-6 also did not provide any citations to Levine with regard to any of the above elements of claim 17.

Additionally, Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest every element of independent claim 25. For example, the Office Action has not cited to portions of Levine that teach, disclose or suggest at least the elements of claim 25: "providing, via the operating system, an application program interface comprising a set of functions" and "storing performance counter information within the performance counter structure." In the Office Action's remarks on page 2 with regard to claim 25, no citations to any passages of Levine for any of the above elements were provided. The Office Action's Response to Arguments on pages 5-6 also did not provide any citations to Levine with regard to any of the above elements of claim 25.

4

Accordingly, the Office Action has not cited to any passages of Levine as disclosing each and every element of independent claims 1, 9, 17 and 25. For at least this reason, the Office Action has not demonstrated that Levine anticipates claims 1, 9, 17 and 25, and independent claims 1, 9, 17 and 25 are allowable under 35 U.S.C. §102(e).

Distinguishing Claims Over Levine:

Furthermore, in addition to the above missing elements, Applicants respectfully submit that the material of independent claims 1, 9, 17 and 25 is patentably distinguishable over Levine. Levine simply does not teach, suggest or disclose each and every element of claims 1, 9, 17 and 25, and therefore does not anticipate claims 1, 9, 17 and 25, as will be discussed below.

Please note that the following discussion is phrased specifically to pertain to claim 1 for clarity and readability. The amendments and discussion, however, apply equally to claims 9, 17 and 25.

Third Element of Claim 1:

For example, the Office Action, page 6 cited to Levine, 9:33-42 as teaching the third element of claim 1 "a performance counter provider application process executed by the processor." Applicants respectfully disagree. Levine, 9:33-42 merely provides a general list of selected performance monitoring functions, each of which may use collected performance data in order to perform its performance monitoring-related responsibility. The items on Levine's list in 9:33-42 are users of the collected performance data, not providers of the collected performance data. None of the items on this general list describe a "performance counter provider application process executed by the processor," as each of the items do not provide or collect performance data but instead *use* the collected performance data.

The present application at the specification, page 2, lines 18-21, describes a performance counter provider as "...generat[ing] the raw counter information. The performance counter provider for an application containing performance counters, consists of

software written specifically for the particular application with which it is associated." The third element of claim 1 requires this counter provider to be an *application process*, which is commonly known as software technology that facilitates exchanging messages or data between two or more different software applications, e.g., the virtual interface between two interworking software functions. The third element further recites that the performance counter provider application process is executed by the processor, thus the performance counter provider application process is a software process. Cited passage Levine, 9:33-42 merely recites a general list of performance monitoring tasks that may use already-collected performance counting data to perform their tasks, but does not teach any of these performance monitoring tasks as collecting or providing the performance data and as being an application process executed by the processor.

Levine does teach a performance counter provider or monitor 50 that generates (rather than uses) counts in its registers (Levine, 7:35-48). However, while Levine teaches his performance monitor 50 being "software-accessible" (Levine, 7:35-38), he does not teach his performance monitor 50 facilitating exchanging of messages or data between two or more different software applications. That is, Levine, 7:35-48 does not teach his performance monitor 50 as being an application or software process, as called for by the third element of claim 1. In fact, Levine, Figure 4 clearly show his performance monitor 50 being implemented in hardware. Therefore, Levine's performance monitor 50 cannot be an application process executed by the processor, as called for by the third element of claim 1.

For at least the above reasons, Applicants respectfully submit Levine does not anticipate the third element of claim 1 "a performance counter provider application process executed by the processor."

Fifth Element of Claim 1:
the application program interface of the operating system

Furthermore, Applicants submit that Levine does not disclose the fifth element of claim 1: "an application program interface of the operating system, comprising a set of functions." As previously discussed, the Office Action did not cite to any passages of Levine

that teach, disclose or suggest this element. In fact, the Office Action was silent as to any passages of Levine with regard to this fifth element.

Indeed, Levine's invention does not teach, suggest or disclose any application program interfaces supported by an operating system. As commonly known in software, an application program interface is an interface between two or more different software applications used to exchange messages and data. Levine's interface between the performance monitor 50 and the software (Levine, Figure 4), however, is a hardware-software interface. In particular, Levine's interface consists of interrupts (Levine Figure 4, references 57 and 260) between hardware and software, and not a set of functions between two different software applications as called for by the fifth element of claim 1. Furthermore, Levine's interface is not "of the operating system." An operating system operates on top of hardware. As the performance monitor 50 of Levine is hardware, the performance monitor 50 and its interface with the software cannot be "of the operating system."

For at least the above reasons, Applicants respectfully submit Levine does not anticipate the fifth element of claim 1 "an application program interface supported by an operating system of the system, comprising a set of functions."

Fifth Element of Claim 1: the counter registration function

The Office Action, page 2 cited to Levine 2:51-64 as teaching the portion of the fifth element of claim 1:

> a counter registration function called by the performance counter provider application process to allocate a performance counter structure within an address space of the performance counter provider application process,
>
> the address space designated by the operating system,
>
> wherein the counter registration function assigns an access function for retrieving performance counter data from the performance counter structure.

Applicants respectfully disagree for at least the following reasons.

7

Levine, 2:51-64 describes code points being inserted into control register locations and generic code points being inserted into other control register locations. The inserted code points then tell the performance monitor when to begin counting a particular event. However, Levine, 2:51-64 does not teach all the above recited limitations. For example, no entity of Levine, 2:51-64 can be construed as "a counter registration function called by the performance counter provider application process." As previously discussed, Levine does not even teach a performance counter provider application process, so Levine cannot teach a performance counter provider application process calling a counter registration function.

Furthermore, Levine, 2:51-64 does not teach "allocate[ing] a performance counter structure within an address space of the performance counter provider application process." As previously discussed, Levine does not teach a performance counter provider application process, so Levine cannot teach an address space of the performance counter provider application process and Levine cannot teach allocating said address space by a counter registration function. In fact, Levine, 2:51-64 does not even describe how his performance counter structures, (i.e., Levine's control register locations) are allocated at all, address space and performance counter provider application process notwithstanding.

Moreover, Levine, 2:51-64 does not teach "assign[ing] an access function for retrieving performance counter data." Again, said passage of Levine merely describes the insertion of code points into control register locations and the inserted code points informing the performance monitor or other counters when to begin counting. Levine, 2:51-64 does not teach an access function for retrieving counted data, much less assigning the access function.

The fifth element of claim 1 provides numerous advantages over Levine. One of the traditional issues associated with rendering performance counter data is overlapping name spaces. Performance data may be suspect or invalid when various different performance counter providers write to common memory spaces, and counter consumers try to retrieve data from common memory spaces.

The present application addresses this issue by having a counter registration function allocate room in the address space of the counter provider application process. As commonly known in the art of software, when an operating system executes an application process, it reserves an address space specifically for that process during the process' lifetime. The performance data is thus linked uniquely to the counter provider application itself as the operating system intrinsically designates address space of the counter provider application to be uniquely owned by that application. Performance data rendering is thus guaranteed to be segregated for that particular counter provider application process, and the data is much more likely to be valid.

The present application provides benefit over Levine by taking advantage of innate operating system characteristics to reallocate memory for performance counters as application processes live and die, thus not only guarding the validity of data in a unique space for the lifetime of a process, but also allowing efficient reuse of memory space without requiring additional hardware registers. Levine, being directed to hardware, simply cannot provide this benefit.

For at least the reasons above, Applicants respectfully submit the cited passage Levine 2:51-64 does not disclose the missing limitations of the fifth element of claim 1:

> a counter registration function called by the performance counter provider application process to allocate a performance counter structure within an address space of the performance counter provider application process,
>
> the address space designated by the operating system,
>
> wherein the counter registration function assigns an access function for retrieving performance counter data from the performance counter structure.

Fifth Element of Claim 1: the counter query

The Office Action, page 2 cited to Levine, 9:32-55 as teaching another portion of the fifth element of claim 1: "a counter query function called by the performance counter

consumer application process to retrieve counter data from the performance counter structure within the address space of the performance counter provider application process by invoking the access function." Applicants respectfully disagree.

Levine, 9:32-55 teaches a list of how collected performance data may be used, e.g., to identify false triggering, to determine the effectiveness of branch unit operations, and others. However, Levine, 9:32-55 does not disclose the specific limitations recited by the fifth element of claim 1. For example, Levine, 9:32-55 does not teach "a counter query function called by the performance counter consumer application process." In fact, as previously discussed, the Office Action did not cite to any passages of Levine as teaching a performance counter consumer application process at all, so Levine cannot teach a counter query function called by the performance counter consumer application process.

Furthermore, Levine, 9:32-55 does not teach the counter query function "retriev[ing] counter data from the performance counter structure within the address space of the performance counter provider application process by invoking the access function." Levine, 9:32-55 does not mention a performance counter provider application process, let alone the address space of the performance counter provider application process, let alone a performance counter structure within the address space by invoking the access function of the previous claim element, the access function allegedly taught (but not found by the Applicant) in Levine, 2:51-64.

Cited passage Levine, 9:32-55 simply does not disclose any mechanism to retrieve data from a performance counter structure within an address space of a counter provider application process, as called for by the fifth element of claim 1. In fact, any performance data retrieved by Levine is retrieved from the hardware PMCs (e.g., PMC1-PMC8 of Levine, 7:39-42) of the hardware performance monitor 50, and not from a performance counter structure defined within an address space of a counter provider application process.

The counter query function of the present application provides benefit over Levine by providing a solution for data retrieval that complements the counter registration function. By

calling a counter query function supported by the operating system, any performance counter consumer application process may easily have valid performance data rendered for its use. The counter query function retrieves the segregated (and therefore, valid) data from the performance data structure that was previously allocated in the address space of the counter provider application process by the counter registration function API. The counter query API to retrieve the segregated performance data used in conjunction with the performance data provision set up by the counter registration function provides a consistent, complementary abstracted interface that the consumer application process may use to obtain valid performance data for further use.

For at least the reasons above, Applicants respectfully submit the cited passage Levine 2:51-64 does not disclose the missing limitations of the fifth element of claim 1 "a counter query function called by the performance counter consumer application process to retrieve counter data from the performance counter structure within the address space of the performance counter provider application process by invoking the access function."

Claims 1, 9, 17 and 25 Conclusion:

As the Office Action has not shown passages of Levine that teach, disclose or suggest each and every element of claim 1, the Office Action has not demonstrated that Levine anticipates each and every element of claim 1. For at least the above reasons, independent claim 1 is thus allowable over Levine under 35 USC §102(e).

Independent claims 9, 17 and 25 each teach features similar to those recited by claim 1, for example, performance counter provider and consumer applications and a set of APIs provided by an operating system including a counter registration API function and a counter query API function. Applicants respectfully submit for at least the reasons similar to independent claim 1, independent claims 9, 17 and 25 are also allowable over Levine under 35 USC §102(e).

11

Claims 2-8, 10-16, 18-24:

Dependent claims 2-8, 10-16 and 18-24 include each and every element of the respective independent claims from which they depend, e.g. independent claims 1, 9 and 17. As Levine does not teach or disclose each and every element of independent claims 1, 9, and 17, Levine also does not teach or disclose each and every element of dependent claims 2-8, 10-16 and 18-24. Accordingly, dependent claims 2-8, 10-16 and 18-24 are not anticipated by Levine and are allowable over Levine under 35 USC §102(e).

Claim 26:

Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest every element of claim 26, and therefore, the Office Action has not established a *prima facie* case of anticipation for claim 26 for at least the reasons discussed below.

First Element of claim 26:

For example, the Office Action has not cited to a portion of Levine that teaches, discloses or suggests the first element of claim 26: "requesting, via an application program interface, the operating system to allocate a memory space within the performance counter provider application process for a performance counter data structure."

The Office Action cited to Levine at col. 6, lines 21-31 as disclosing this first element of claim 26. This passage, however, does not disclose requesting an operating system, via an application program interface, to allocate a memory space within a performance counter application process. Levine, col. 6, line 21-31 is merely a general description of how instructions in a processor are sequenced and dispatched to appropriate execution units in the processor and does not discuss performance monitoring. Indeed, at this point of Levine's detailed description, performance counting has not even been introduced. A performance counter application process is not described in this passage, an application program interface (i.e., an interface between software entities) is not described in this passage, and a request to the operating system is also not described in this passage.

Furthermore, Levine, 6:21-31 also does not describe an allocation of memory space within the address space of a performance counter provider application process. Not only does Levine, 6:21-31 not mention allocation of memory space within the address space of a performance counter provider application process, said passage does not mention the address space of the performance counter provider application process, or even a performance counter provider application process. Levine, 6:21-31 merely describes sequencing and dispatching instructions to execution units in a processor.

For at least the reasons above, Applicants respectfully submit the cited passage Levine, 6:21-31 does not disclose the first element of claim 26 "requesting, via an application program interface, the operating system to allocate a memory space within the performance counter provider application process for a performance counter data structure."

Third element of claim 26:

Additionally, Applicants submit the Office Action has not cited to a portion of Levine that teaches, discloses or suggests the third element of claim 26: "providing access by a registered callback function, invoked by a call to the application program interface, to the memory space containing the performance counter data structure." The Office Action cited to Levine, 4:64-67, 5:1-9 and 9:17-25 as teaching the third element of claim 26. Applicants respectfully disagree.

The Office Action cited to Levine, 4:64-67 and 5:1-9 as teaching "providing access by a registered call back function." Levine, 4:64-67 and 5:1-9 generally describes storing information in a rename buffer 34 and associated the stored information with a storage location (e.g., one of GPRs 32, CA register 42 or one of the FPRs 36). A sequencer unit 18 directs the copying of the stored information from the rename buffer into the storage location.

The Office Action cited to Levine, 9:17-25 as teaching "invoked by a call to the application program interface, to the memory space containing the performance counter data structure." Levine, 9:17-25 generally describes adjusting values of performance monitor counts.

13

This combination of cited Levine passages, however, does not teach the third element of claim 26. First, Applicants are unclear as to which entity in Levine, 4:64-67 and 5:1-9 the Office Action is alleging to teach the registered callback function. Levine, 4:64-67 and 5:1-9 do not describe any functions at all, let alone registered callback functions. The only verbs or actions (and it is arguable if these verbs or actions are or are not the "function" of the claim) that Applicants find in said passage are an action of storing information into a rename buffer, an action of copying the information from the rename buffer into a storage location, and an action of directing the copying. None of these three actions, however, are disclosed by Levine, 4:64-67 and 5:1-9 to be either "registered" or a "callback function," nor are they disclosed to provide access to the rename buffer or storage location "by a registered callback function."

Furthermore, these three actions are taught in Levine, 4:64-67 and 5:1-9 as operating on either the rename buffer 34 or the storage locations (32, 36, 42), but the three actions are not taught in said passage as operating on any storage location that contains performance data counts. The third element of claim 26 requires the registered callback function to operate on the memory space containing the performance counter data. The memory space containing the performance counter data was alleged by the Office Action's citation to Levine, 9:17-25 to be the location where the performance monitor counts are stored. However, the storage location of the performance monitor counts of Levine is not the rename buffer 34, the GPRs 32, the FPRs 36, or the CA 42. Instead, the storage location of performance monitor counts is the counters PMC1-PMC8 (Levine 7:39-42, reference 51 of Fig. 4). Thus, the combined cited passages of Levine 4:64-67, 5:1-9 and 4:64-67 do not teach the third element of claim 26.

For at least the reasons above, Applicants respectfully submit the cited passages Levine, 4:64-67, 5:1-9 and 9:17-25 do not disclose the third element of claim 26 "providing access by a registered callback function, invoked by a call to the application program interface, to the memory space containing the performance counter data structure."

<u>Conclusion for claim 26:</u>

For at least the above reasons, Applicants submit Levine does not disclose each and every element of claim 26. Accordingly, Levine does not anticipate claim 26 and claim 26 is allowable under 35 USC §102(e).

Claim 27:

Applicants respectfully submit that the Office Action has not cited to portions of Levine that teach, disclose or suggest every element of claim 27.

For example, the Office Action did not cite to any passage of Levine as teaching the element of claim 27 "an operating system executed by the processor."

Further, the Office Action cited to Levine, 9:43-55 as teaching the elements of claim 27:

> a performance counter provider application process executed by the processor;
>
> a performance counter consumer application process executed by the processor; and
>
> an operating system performance counter application program interface comprising a first set of functions callable by the performance counter provider application process to register a corresponding performance counter provider in a repository...

Applicants respectfully disagree. Levine, 9:43-55 generally describes analyzing collected performance data after the completion of performance monitoring.   Applicants do not find in Levine, 9:43-55 any teaching of "a performance counter provider application process executed by the processor" or "a performance counter consumer application process executed by the processor" In fact, Levine, 9:43-55 does not even mention application processes executed by the processor at all.

Levine, 9:43-55 also does not teach an operating system performance counter API. Indeed, Levine, 9:43-55 does not even mention APIs at all, let alone operating system APIs or an operating system performance counter API.

Levine, 9:43-55 also does not teach registering a corresponding performance counter provider in a repository. The claim element calls for registering or setting up a specific performance counter via a set of first functions provided by an operating system API to aid in the collection of performance data. However, all of Levine, 9:43-55 does not occur during performance data collection or even during a preparation for performance data collection. Rather, Levine, 9:43-55 discusses how performance data may be analyzed after the collection has completed.

Still further, the Office Action cited to Levine, 2:51-64 as teaching "a second set of functions for serving requests originating from the performance counter consumer application process to enumerate and access the performance counter provider application process." Applicants respectfully disagree. Levine, 2:51-64, as previously discussed, describes inserting code points into control location registers where the inserted code points notify the performance monitor to begin counting for a particular control location register.

As the Office Action was silent on which entity of Levine the Office Action alleges to be the "performance counter consumer application process" and the "performance counter provider application process," Applicants are unable to assess whether or not Levine, 2:51-64 teaches "requests originating from the performance counter consumer application" and whether or not these requests pertain "to... the performance counter provider application process," as called for by this element of claim 27. Nonetheless, cited passage Levine, 2:51-64 still does not teach a second set of functions of an operating system performance counter API, nor does said passage teach serving requests of any sort or enumerating any counter provider application processes.

As discussed earlier, Levine's invention is directed towards collecting raw performance data using registers in a hardware performance monitor. Additional desired performance counts require additional hardware registers to be added (Levine, col. 7, line 65-col. 8, line 3) thus adding extra cost and expense. The present application provides benefit over Levine by taking advantage of innate operating system characteristics to reallocate memory for performance counters as application processes live and die, thus not only guarding the validity of data in a unique space for the lifetime of a process, but also allowing efficient reuse of memory space without requiring additional hardware registers.

For at least the above reasons, Applicants submit the Office Action has not demonstrated each and every element of claim 27 is taught by Levine. Accordingly, Levine does not anticipate claim 27 and claim 27 is allowable under 35 USC §102(e).

CONCLUSION

In view of the above, each of the claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to withdraw the outstanding rejection of the claims and to pass this application to issue.

Dated: February 11, 2009   Respectfully submitted,

                                                        By__/W. J. Kramer #46,229/_____
                                                        William J. Kramer
                                                           Registration No.: 46,229
                                                        MARSHALL, GERSTEIN & BORUN LLP
                                                        233 S. Wacker Drive, Suite 6300
                                                        Sears Tower
                                                        Chicago, Illinois 60606-6357
                                                        (312) 474-6300
                                                        Attorney for Applicant